I. CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method for graphically representing object oriented programming logic, the method comprising the steps of:

(1) providing, via a specially programmed computer, a graphical user interface (GUI) presenting a plurality of different symbols for use in a diagram of object oriented programming logic, each different symbol representing a different type of object in object oriented programming;

(2) ~~selecting~~ receiving a selection, via an input device of the specially programmed computer, of an object as a main object of the logic of a program to be represented in the diagram; and

in response to the received selection within the GUI using said specially programmed computer:

(3) drawing, via the GUI, ~~, using said specially programmed computer,~~ a symbol corresponding to the main object of the program and labeling the symbol with a label descriptive of the object's features so that it is distinguishable from other symbols of [[the]] a same object type;

(4) for each object assigned to or defined within the main object of the program, drawing, via the GUI, ~~, using said specially programmed computer,~~ a symbol corresponding to that object and labeling the symbol with a label descriptive of the object's features; [[and]]

(5) drawing, via the GUI, ~~, using said specially programmed computer,~~ a line between each symbol drawn in step (4) and another symbol in the graphical representation corresponding to [[an]] another object to which said object corresponding to said symbol drawn in step (4) is assigned or within which the another object is defined; and

(6) graphically denoting, via the GUI, the symbol in the diagram corresponding to the main object of the program so as to distinguish it from other symbols in the diagram by drawing an additional symbol around the symbol corresponding to the main object of the program.

2. (Currently Amended) The method of claim 1 further comprising the step of:

[[(6)]] providing, via the GUI, a plurality of additional different symbols for use in the diagram, each of the additional different symbols representing a different object oriented programming element type other than an object.

3. (Cancelled)

4. (Currently Amended) The method of claim [[3]] 1 wherein step [[(7)]] (6) comprises drawing a circle completely enclosing the symbol of the main object of the program within the GUI.

5. (Original) The method of claim 1 wherein the labels comprise text.

6. (Currently Amended) The method of claim 1 wherein step (5) comprises drawing, via the GUI, the line between the symbol corresponding to the object defined in step (4) and the another symbol corresponding to the another object it is most directly assigned to or is most directly defined within.

7. (Previously Presented) The method of claim 1 wherein the method is used to document pre-existing object oriented programming software.

8. (Original) The method of claim 1 wherein the method is used to prepare a program specification.

9. (Currently Amended) The method of claim 1 further comprising the step of:

(8) repeating steps (1) - (5) to prepare a plurality of separate diagrams corresponding to separate parts of an overall application and wherein a first object is the main object of the program appearing in at least a first one of the diagrams and is not a main object of the program appearing in at least a second one of the diagrams.

10. (Original) The method of claim 9 wherein the second diagram does not disclose objects assigned to and defined within the first object and the first diagram does disclose objects assigned to and defined within the first object.

11. (Original) The method of claim 10 wherein the second diagram is an application-level representation disclosing an overall software system.

12. (Previously Presented) The method of claim 10 wherein the label for the symbol corresponding to the first object in the second diagram identifies the first diagram as disclosing further details of the first object.

13. (Previously Presented) The method of claim 1 wherein the symbols representing different object types include at least the five following symbols:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and

a fifth symbol for representing objects that are method type objects.

14. (Previously Presented) The method of claim 2 wherein the symbols representing different object types include at least the following five symbols:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and

a fifth symbol for representing objects that are method type objects;

and wherein the symbols representing additional program elements include:

a sixth symbol for representing data transfer;

a seventh symbol for representing databases;

an eighth symbol for representing remote links; and

a ninth symbol for representing inheritance.

15. (Original) The method of claim 14 wherein the sixth, eighth, and ninth symbols are drawn connecting two other object symbols.

16. (Currently Amended) The method of claim 14 wherein the symbols representing different object types further include at least the following five additional symbols [[;]] :

a tenth symbol for representing objects that are menu type objects;

[[a]] an eleventh symbol for representing objects that are frame type objects;

a twelfth symbol for representing objects that are button type objects;

a thirteenth symbol for representing objects that are data structure type objects; and

a fourteenth symbol for representing objects that are not one of the other object types.

17. (Original) The method of claim 13 further comprising the step of:

(9) providing in a separate document a description of the logic to be performed responsive to an event script.

18. (Original) The method of claim 13 wherein the fourth symbol representing event script type objects is drawn connected to another object that directly executes the event script corresponding to the event script symbol.

19. (Currently Amended) The method of claim 13 wherein the fifth symbol representing method type objects is drawn connected to the main object of the program within the diagram and represents that the object is available within [[that]] the main object of the program and does not represent that the main object of the program invokes it.

20. (Currently Amended) The method of claim 1 wherein step (1) comprises providing a graphical user interface in which a user is presented with a pallet containing the symbols and wherein receiving the selection via the input device, in response to which steps (3) and (4) ~~comprise~~ are performed, comprises detecting dragging and dropping mouse actions associated with the symbols from the pallet into a work area.

21. (Currently Amended) The method of claim 1 wherein step (1) comprises providing a graphical user interface in which a user is presented with a pallet containing the symbols and wherein receiving the selection via the input device, in response to which steps (3) and (4) ~~comprise~~ are performed, comprises detecting dragging and dropping mouse actions associated with the symbols from the pallet into a work area, and wherein the labels comprise text and further wherein at least some of the text labels are hidden text ~~that can be~~ made to appear in the graphical representation via an input action taken by a user.

22. (Currently Amended) A computer readable product embodied on computer readable media readable by a computing device for enabling a user to generate a graphical representation of object oriented programming logic, the product comprising:

first computer executable instructions that provide, via the computing device, a graphical user interface (GUI) in which a user is presented with a plurality of different symbols for use in developing a graphical representation of object oriented programming logic, each different symbol representing a different type of object in object oriented programming;

second computer executable instructions that ~~enable the user, to select~~ cause the computing device to receive a selection of one and only one object in the diagram as a main object of the logic of a program represented in the diagram; and

in response to the received selection, via the computing device, within the GUI:

third computer executable instructions that ~~enable the user to~~ cause the computing device to draw a symbol corresponding to the main object of the program and label the symbol with a label descriptive of the object's features so that it is distinguishable from other symbols of [[the]] a same object type;

fourth computer executable instructions that ~~enable the user to~~ cause the computing device to detect drag and drop mouse actions associated with symbols corresponding to object-oriented programming objects into a workspace and label the symbols with a label descriptive of the corresponding object's features;

fifth computer executable instructions that ~~enable the user to~~ cause the computing device to draw a line between each symbol dropped in the workspace by the fourth computer executable instructions and another symbol in the workspace corresponding to [[an]] another object to which [[it]] said object corresponding to each symbol dropped in the workspace by the fourth computer executable instructions is assigned or within which [[it]] the another object is defined; and

sixth computer executable instructions that ~~enable the user to~~ cause the computing device to graphically denote the main object of the program in the diagram by drawing ~~another~~ an additional symbol around the symbol corresponding to the main object of the program.

23. (Cancelled)

24. (Cancelled)

25. (Currently Amended) The computer readable product of claim 22 wherein the sixth computer executable instructions comprise instructions ~~enabling the user to~~ that cause the computing device to enclose the symbol corresponding to the one and only one object within a circle.

26. (Original) The computer readable product of claim 22 wherein the labels are text labels.

27. (Currently Amended) The computer readable product of claim 22 further comprising:

seventh computer readable instructions that ~~enable the user to~~ cause the computing device to prepare a plurality of the diagrams corresponding to separate parts of an overall application and further comprising computer readable instructions ~~for enabling the user to~~ that cause the computing device to specify relationships between individual ones of the diagrams.

28. (Currently Amended) The computer readable product of claim 27 wherein the seventh computer readable instructions comprise instructions that ~~enable the user to~~ cause the computing device to include references associated with symbols in one diagram identifying at least one other diagram within which the object represented by that symbol also appears.

29. (Currently Amended) The computer readable product of claim 28 wherein the seventh computer readable instructions comprise instructions that ~~enable the user to~~ cause the computing device to specify in a first one of the diagrams the nature of the relationship of the representation

of the object in the first diagram relative to the representation of the object in a second diagram, wherein the relationship between the object as represented in the first and second diagrams is selected from the group comprising: (1) the second diagram discloses additional details about the object in the first diagram; (2) the second diagram shows the object in a more abstract context than the first diagram; and (3) the object is the main object of the program of the second diagram.

30. (Original) The computer readable product of claim 22 wherein the symbols representing different object types include:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and

a fifth symbol for representing objects that are method type objects.

31. (Currently Amended) The computer readable product of claim 24 wherein the symbols representing different object types include:

a first symbol for representing objects that are application type objects;

a second symbol for representing objects that are window type objects;

a third symbol for representing objects that are class type objects;

a fourth symbol for representing objects that are event script type objects; and a fifth symbol for representing objects that are method type objects [[.]] ;

and wherein the additional symbols representing additional program elements include:

a sixth symbol for representing data transfers;

a seventh symbol for representing databases;

an eighth symbol for representing remote links; and

a ninth symbol for representing inheritance.

32. (Currently Amended) The computer readable product of claim 31 further comprising:

eighth computer executable instructions that restrict the [[user]] computing device to using the sixth, eighth, and ninth symbols to connect two other object symbols.

33. (Currently Amended) The computer readable product of claim 30 wherein the symbols representing different object types further include [[;]] :

a tenth symbol for representing objects that are menu type objects;

an eleventh symbol for representing objects that are frame type objects;

a twelfth symbol for representing objects that are button type objects;

a thirteenth symbol for representing objects that are data structure type objects; and

a fourteenth symbol for representing objects that are not one of the other object types.

34. (Currently Amended) The computer readable product of claim 30 further comprising:

ninth computer executable instructions that ~~enable the user to~~ cause the computing device to provide in a separate document a description of the logic to be performed responsive to an event script.

35. (Currently Amended) The computer readable product of claim 22 further comprising:

tenth computer executable instructions that ~~enable the user to~~ cause the computing device to insert hidden text associated with symbols in the workspace ~~that can be~~ made to appear in the workspace responsive to an input action taken by a user.